Optimization of fractal : function using genetic algorithms

Jacques Lévy Véhel, Evelyne Lutton

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Optimization of Fractal Functions using Genetic Algorithms

Jacques LEVY-VEHEL
Evelyne LUTTON

N° 1941
Juin 1993

PROGRAMME 4

Robotique, image
et
vision

Optimization of Fractal Functions using Genetic Algorithms

Optimisation de Fonctions Fractales par Algorithmes Génétiques

Jacques LEVY-VEHEL, Evelyne LUTTON
INRIA - Rocquencourt
B.P. 105, 78153 LE CHESNAY Cedex, France
Tel : 33 1 39 63 55 23 - Fax : 33 1 39 63 53 30
email : evelyne.lutton@inria.fr, jlv@bora.inria.fr

Abstract

In this work, we investigate the difficult problem of the optimization of fractal functions. We first derive some relations between the local scaling exponents of the functions, the sampling rate, and the accuracy of the localization of the optimum, both in the domain and the range of the functions. We then apply these ideas to the resolution of the inverse problem for Iterated Function System (IFS) using a Genetic Algorithm. In the conditions of study (2D problem for sets), the optimization process yields the optimum with a good precision and within a tractable computing time.

Keywords : Fractals, Genetic Algorithms, Optimization, IFS.

Résumé

Dans cette étude, nous abordons le difficile problème de l'optimisation de fonctions fractales. Nous établissons tout d'abord des relations entre les exposants d'échelle locaux des fonctions, le taux d'échantillonnage et la précision sur la localisation de l'optimum. Nous appliquons ensuite ces résultats à la résolution du problème inverse pour les Systèmes de Fonctions Itérées (IFS), en employant un algortihme génétique. Sous les conditions étudiées (problème bidimensionnel pour les ensembles), le processus d'optimisation fournit une estimation de la solution optimale avec une bonne précision et en un temps CPU acceptable.

Mots-clés : *Fractales, Algorithmes Génétiques, Optimisation, IFS.*

1 INTRODUCTION

It sometimes happens that solving a particular problem amounts to find the minimum of a function over a given space. One then talk of an optimization problem. When the function has a certain type of regularity, a number of methods exists, mostly based on gradient or generalized gradient computations (see for instance [7]).

Generalized gradient methods work well when :

- some sort of gradient can be defined and computed at any point of the space of solutions (for instance, directional derivatives),
- the function does not have too many local minima, or the value taken by the function at these minima is significantly greater than the value at the absolute minimum.

Fractal functions, as the Weierstrass function displayed on figure 1, do not satisfy these requirements : they are nowhere differentiable, no directional derivatives exists, and they possess infinitely many local minima.

For such and other very irregular functions, different methods have to be used for optimization. Most of them are based on stochastic schemes.

One of the most known stochastic algorithms is Simulated Annealing. It is a powerful technique for finding the global minimum of a function when a great number of parameters have to be taken into account. It is based upon an analogy with the annealing of solids, where a material is heated to a high temperature, then very slowly cooled in order to let the system reach its ground energy. The delicate point is to lower the temperature T no too rapidly, thus avoiding local minima. Application to other optimization problems is done by generalizing the states of the physical system to some defined states of the system being optimized, and generalizing the temperature to a control parameter for the optimization process ; most of the time, the Metropolis algorithm is used : at "temperature" T, the jump from a state of energy E to a state of energy E' is made with probability one if E' is lower than E and with a probability proportional to (E-E')/T if not ([1, 20]). The main drawback of Simulated Annealing is the computational time : the optimal solution is guaranteed only if the temperature is lowered at a logarithmic rate([11]), implying a huge number of iterations. Most of the time, a linear rate is used to obtain affordable converging times, but, for certain very wild functions, the logarithmic rate has to be used. However,Simulated Annealing have been used with some success for the optimization of fractal functions (see [18]).

In this work, we investigate the use of another recently introduced method for stochastic optimization, namely Genetic Algorithms. We first state some basic facts about how the optimization of a fractal function should be conducted and which type of results one may expect, using extensively the (possibly local) self similar behavior of the function (section 2). In section 3, we recall the main aspects of genetic algorithms. In section 4, we validate the use of such algorithms by applying them to well known functions, such as the Weierstrass one. We then present an application to the resolution of the 2D inverse problem for IFS (section 5) and sum up our results in section 6, proposing various desirable extensions.

2 OPTIMIZATION OF A FRACTAL FUNCTION

In this section, and throughout the paper, we only consider discrete optimization methods : that is, the solution x that minimizes f is searched over a discrete subset of the domain of f. This corresponds to the practical case where the variable has to be coded on a fixed and finite number of bits.

We restrict ourselves to the case where the functions f has, at each point, a scaling behavior. This means that :

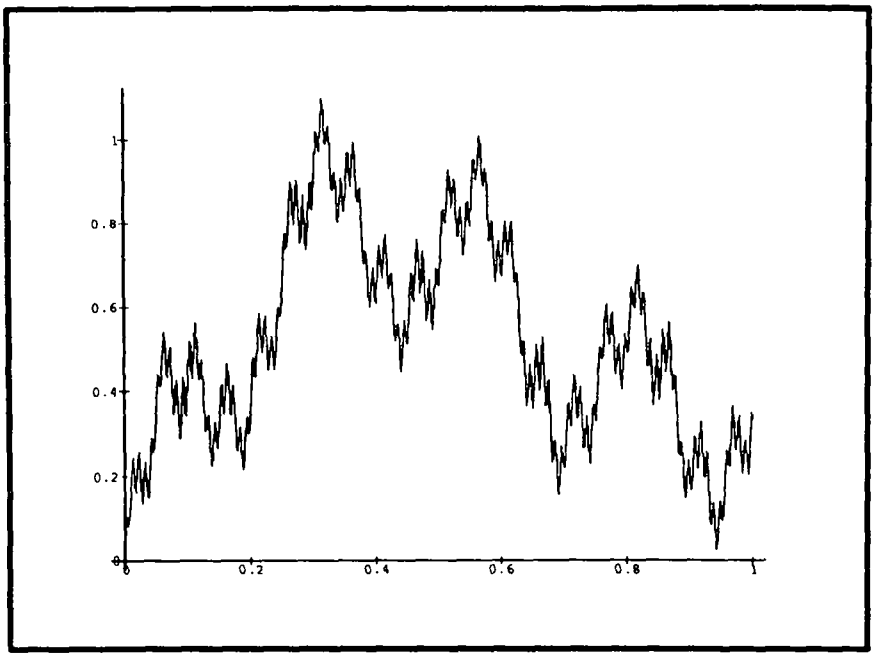

Figure 1: Weierstrass function of dimension 1.5.

$$\forall x, u \in I^2, \forall \lambda \in R, f(x + \lambda u) - f(x) \sim |\lambda|^h f(u) \quad when \quad \lambda \to 0$$

where :

- f is a function defined on $I \subset E$, E being a subset of R^n, taking values on F, a subset of R, and u is a normalized vector,
- $g(\lambda) \sim l(\lambda)$ *when* $\lambda \to 0$ means that : $\lim_{\lambda \to 0} \frac{||g(\lambda)||}{||l(\lambda)||} = k \in R$ (i.e. the limit exists and is finite).

Our aim is to derive some general bounds on the precision of the optimization, both in the domain and range of f, related to the discretization of the domain of the function. In other words we want to answer the questions :

- at a given sampling rate, what is the maximum error that can be made on the location of the absolute minimum, and on the corresponding value of the function ?
- if this error is too large, by how much do we have to refine the sampling to obtain a given precision ?

For the sake of clarity, we first consider real-valued functions whose domain is $]0, 1[$.

Assume that the variable x is coded with n bits. This correspond to a sampling of the interval $]0, 1[$ at frequency $F_0 = 2^n$

If g is a smooth function whose frequency spectrum is included in $[0, F]$, we know by Shannon's theorem that if F_0 is greater than F, then no information is lost by sampling at frequency F_0. Thus

it is theoretically possible to find the global minimum of g on the discretized space, with accuracy $\epsilon = 2^{-n}$ both for x and g(x). In other words :

$$\begin{aligned} x_{min} &\in]\hat{x} - \epsilon, \hat{x} + \epsilon[\\ g(x_{min}) &\in]g(\hat{x}) - \epsilon|g'(\hat{x})|, g(\hat{x}) + \epsilon|g'(\hat{x})|[\quad when \ \epsilon \to 0 \end{aligned}$$

where x_{min} is the real minimum, and $\hat{x}$ the best value found by the optimization algorithm.

However in our case, f is a fractal function, and its frequency spectrum is unbounded : any sampling implies a loss of information. No matter how large n is, some variations of the function will be lost.

Many fractal functions have an $\frac{1}{f^\beta}$ spectrum, with $\beta \in]1, 3[$. For instance, fractional Brownian motion of exponent H has an $\frac{1}{f^{2H+1}}$ spectrum, for $H \in]0, 1[$.

The Weierstrass function defined as :

$$W_{b,s}(x) = \sum_{i=1}^{\infty} b^{i(s-2)} \sin(b^i x)$$

with $b > 1$ and $1 < s < 2$, also has an $\frac{1}{f^\beta}$ spectrum, with $\beta = 5 - 2s$

If we sample these functions at frequency F_0, we can evaluate how much information we have lost, by computing the integral :

$$\int_{F_0}^{\infty} \frac{1}{f^\beta} df$$

which converges, since $\beta > 1$.

Further analysis of this quantity allows to assert, in a probabilistic way, the quality of the optimization ; these results will be presented in a forthcoming paper. Here we use another path to heuristically derive some bounds on the precision.

Let $x_0, x_1, ...x_{2^n}$ be the sampling points at frequency F_0. Suppose that the optimization method returns x_m as the global minimum. Assume further that the point where f reaches its second minimum is $x_{m'}$, also computed by the optimization algorithm :

$$\begin{aligned} \forall i \in 0, 1, ..., 2^n, \ f(x_m) \leq f(x_i) \\ \forall i \in 0, 1, ..., 2^n, i \neq m, \ f(x_{m'}) \leq f(x_i) \end{aligned}$$

Using some theoretical considerations, or some measures performed on the signal, we have access to the value of h with a precision of $\delta > 0$ ($\delta < h$). This means :

$$\hat{h} \in [h - \delta, h + \delta]$$

where $\hat{h}$ is the estimated value.

From the scaling property of f, we know that :

$$\forall x \in]x_{m'} - \epsilon, x_{m'} + \epsilon[, f(x) - f(x_{m'}) \simeq k\eta^h$$

where $\eta = |x - x_{m'}|$, provided that ϵ is small enough.

Let $y_m = f(x_m), y_{m'} = f(x_{m'}), y = f(x)$. We have, for $x \in]x_{m'} - \epsilon, x_{m'} + \epsilon[$:

$$y \simeq y_{m'} + k\eta^h$$

Now we need to find some bounds for k. Since η is positive, we may define :

$$q = \frac{ln|k|}{ln(\eta)} \qquad or \ : \qquad k = \pm\eta^q$$

This implies :

$$f(x) - f(x_{m'}) \simeq \pm\eta^{q+h}$$

k is thus related to the incertitude on h, since measuring $f(x) - f(x_{m'})$ will give us an evaluation of $\hat{h}$:

$$h - \delta \leq q + h \leq h + \delta$$

$$or : \qquad -\delta \leq q \leq \delta$$

We then have a lower bound for y, using the facts that $\eta < \epsilon$ and $h + \delta \leq 1$:

$$y \geq y_{m'} - \epsilon^{h-\delta} \qquad if \quad \epsilon \ll 1$$

Thus, we alway have :

$$\forall x \in]x_{m'} - \epsilon, x_{m'} + \epsilon[, \; y = f(x) \geq y_{m'} - \epsilon^{h-\delta}$$

The "worst" case we have to consider is then the one when there actually exists $x_0 \in]x_{m'} - \epsilon, x_{m'} + \epsilon[$ such that :

$$y = f(x_0) = y_{m'} - \epsilon^{h-\delta}$$

If it happens that : $y_{m'} - y_m < \epsilon^{h-\delta}$,
then : $y_0 = f(x_0) < y_m = f(x_m)$, causing a completely false localization of the minimum.
So far, we have thus obtained that :

- if a fractal signal is sampled at frequency 2^n on $]0, 1[$, if the self-similarity exponent h is known with accuracy δ, and if we write :

  $$f(x) - f(x_i) = k(x)\epsilon^h \qquad for \qquad x \in]x_i - \epsilon, x_i + \epsilon[$$

  with $\epsilon = 2^{-n}$, and x_i is a sampling point, then :

  $$2^{-n\delta} \leq k(x) \leq 2^{n\delta}$$

- if x_m is the global minimum, $x_{m'}$ the second minimum both with respect to computation only on the sampling points, and if :

  $$f(x_{m'}) - f(x_m) < \epsilon^{h-\delta}$$

  then there might exist $x_0 \in]x_{m'} - \epsilon, x_{m'} + \epsilon[$ such that : $f(x_0) < f(x_m)$

This mean that, contrary to the case of signals with limited spectrum, where the minimum may be localized to within an error of the order of the sampling period, the error on the localization of the minimum on a fractal signal may be arbitrarily large, no matter how precisely the sampling is done.
However, if :

$$f(x_{m'}) - f(x_m) > \epsilon^{h-\delta} \tag{1}$$

then we know that, as in the case of regular functions, the localization is accurate, in the sense that the true minimum x^* belongs to $]x_m - \epsilon, x_m + \epsilon[$.

In that case, we may also evaluate the accuracy of the value of the minimum :

$$x^* \in]x_m - \epsilon, x_m + \epsilon[\quad \Rightarrow f(x_m) - \epsilon^{h-\delta} \leq f(x^*) \leq f(x_m) + \epsilon^{h-\delta} \tag{2}$$

Thus the error on $f(x^*)$ is lower than $\epsilon^{h-\delta}$.

Example : suppose that we are trying to compute the minimum of a fractal function on $]0, 1[$, using a discretization on n = 10 bits, and that we know that $h = 0.5 \pm 0.1$. Then if the difference between $f(x_m)$ and $f(x_{m'})$ is greater than 2^{-4}, we know that we have a 2^{-10} accuracy on the position of the minimum, and a 2^{-4} accuracy on its value.

As a by-product of this analysis, we have that, to evaluate the performances of an optimization algorithm on a fractal function, we have to compare the number of trials or visited sites directly to the number of sampling points, since the sampling never exhausts the total information carried by the signal.

We can also define a general method for optimization of a fractal function f :

- choose a sampling frequency F,
- run the optimization algorithm, and keep the two best values in the domain and range of f,
- evaluate the local exponent h at these points,
- if (1) is not fulfilled, increase the sampling frequency and run the algorithm again,
- if (1) is fulfilled, then x^* is known with accuracy given by the sampling period, and $f(x^*)$ with accuracy given by (2).

3 GENETIC ALGORITHMS

We will use Genetic Algorithms as stochastic optimization methods, but it must be noticed that they also have other fields of applications, as for instance in image processing, automatic programming, graph theory, etc ... (see [13, 21, 2, 23, 16, 12, 8]).

The benefit of using Genetic Algorithms to optimize fractal functions is that they perform a stochastic search over a large searching space, by making evolve together a set of solution, instead of using a unique solution as in the Simulated Annealing scheme.

We use here a sequential implementation of a Genetic Algorithm, but notice also that G.A. have the great advantage to be easily parallelized.

The specificity of genetic algorithms is that they try to copy simple natural evolution schemes. John Holland is largely recognized as the founder of the field of Genetic Algorithms [14], he integrated and elaborated two themes that had persistently recurred in his research : the ability of simple representations (bit strings) to encode complicated structures, and the power of simple transformations to improve such structures. Holland showed that with the proper control structure, rapid improvements of bit strings could be made to "evolve" as population of animals do. An important formal result stressed by Holland was that even in large and complicated search spaces, given certain conditions on the problem domain, genetic algorithms would tend to converge on solutions that are globally optimal or nearly so.

In natural evolution, the problem each species faces is the one of searching for beneficial adaptations to a complicated and changing environment. The "knowledge" that each species has gained is embodied in the makeup of the chromosomes of its members. The operations that alter this chromosomal makeup are applied when parents reproduce ; among them are random mutation, inversion

of chromosomal material, and crossover - exchange of chromosomal material between two parents' chromosomes (see figures 2 and 3). This feature of natural evolution - the ability of a population of chromosomes to explore its search space in parallel and combine the best findings through crossover - is exploited when genetic algorithms are used.

A genetic algorithm (G.A.) to solve a problem is commonly described as having 5 components (see [9]) :

1. *a chromosomal representation of solutions to the problem,*
2. *a way to create an initial population of solutions,*
3. *an evaluation function that plays the role of the environment, rating solutions in term of their "fitness",*
4. *genetic operators that alter the composition of children during reproduction,*
5. *values for the parameters that the genetic algorithm uses (populations size, probabilities of applying genetic operators, etc ...)*

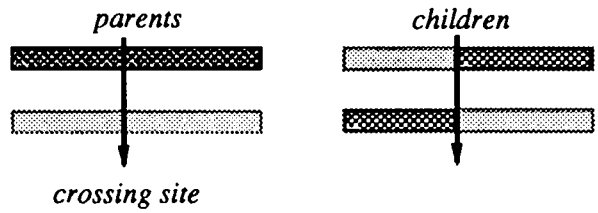

Figure 2: Crossing Over

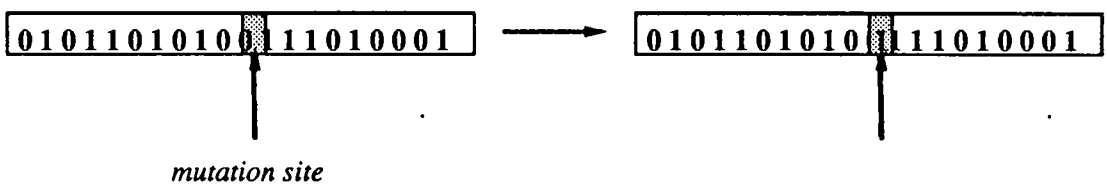

Figure 3: Mutation

In our applications to fractal functions optimization, we have made some choices concerning these points, in order to perform an efficient statistical optimization. Among the many possible variations of the structure of a G.A., we implemented a classical scheme, where the population size of each generation is constant, and the initial population is randomly generated on the search space.

- **Genes**

  We have to handle continuous search spaces, that we discretize, with N bits for each parameter, thus defining the precision we use to analyse the searching space (see section 2).

- **Genetic operators**

  We use the one point cross over, and mutation (as described in figures 2 and 3).

  Our selection scheme is also a simple one : the roulette wheel selection (see figure 4) where we select two parents according to their fitness. The best ones are more often selected than the others. To implement it, we use a uniform random selection on a sort of wheel, where, for each individual of the population, sectors proportional to their fitness are defined.

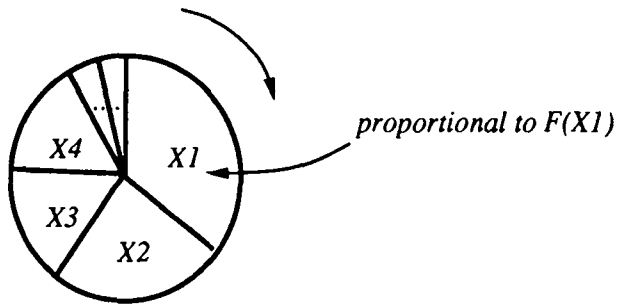

Figure 4: Roulette wheel selection

In fact we use a transform of the fitness function in order to avoid the domination of some super-individuals (that has a great fitness, and are almost always selected as parent). We linearly scale the fitness so that the best individual has a fitness equal to S times the mean fitness of the population (for precisions, see [12], S is generally between 1.2 and 2). Note that the scaling factors are computed at each generation.

- **Parameters**

  These parameters are very important, because they severely influences the convergence speed :

  - *population size and chromosome size :* the population size is taken equal to the total number of bits of a chromosome, and the chromosome size is chosen with help of the considerations of the section 2,
  - *selection probability :* it is relatively high, because this operator is the most important factor which guides the exploration of the searching space ; it is taken equal to 0.7,
  - *mutation probability :* it is very low, and is chosen to be such that on average 1 bit per 1000 is changed.

- **Stopping criterion**

  The problem of stopping the search is problematic, and few authors have proposed a solution. Indeed, when can we assume that the algorithm will no more evolve ? The most simple solution is to stop at a fixed number of generations. But, according to the mechanism of the G.A., we

can say that the population tends to converge near a global minima. We thus propose to stop the search when the difference between the maximum and mean fitness of the population is lower than a fixed threshold.

Remark : in order to use the results of section 2, we need to find a second local minimum. We perform that by running again the G.A. but on the space of solutions without the neighborhood of the first optimum. Indeed, it is not possible to find the first and the second optimum in an unique run, because the G.A. tends to concentrate the solutions around the global minimum.

4 EXPERIMENTS ON KNOWN FUNCTIONS

In order to verify experimentally the ideas presented in section 2 and 3, we perform some tests on simple functions, namely Weierstrass functions. On figure 1 and 5 are shown Weierstrass functions with h = 0.5 and 0.3 respectively, which corresponds to fractal dimensions of 1.5 and 1.7. This is a simple case since h is constant and $\delta = 0$ for each function. The range of x is [0,1], and b is taken equal to 5.

1. $\boxed{\text{h = 0.5}}$ Minimization of $W_{5,1.5}$:

   According to section 2, the test for accuracy is :

   $$f(x_{m'}) - f(x_m) > \epsilon^{0.5} = 2^{-n/2} \tag{3}$$

   We start with an 8-bit discretization, and run the GA. The estimations are :

   $$\begin{aligned} x_m &= 0. & f(x_m) &= 0. \\ x'_{m'} &= 0.941 & f(x_{m'}) &= 0.065 \end{aligned}$$

   We compute :

   $$f(x_{m'}) - f(x_m) = 0.065 > 2^{-4} = 0.062$$

   Since the test is verified, we expect good localization. In fact, as one can verify it on figure 1, the G.A. has found the right minimum.

2. $\boxed{\text{h = 0.3}}$ Minimization of $W_{5,1.7}$:

   This time, the test reads : $f(x_{m'}) - f(x_m) > 2^{-0.3*n}$. This condition is harder to fulfill which is normal since the second curve is much more irregular than the previous one (or, equivalently, the decreasing of the power spectrum is slower).

   The G.A. gives, with an 8 bits discretization :

   $$\begin{aligned} x_m &= 0.941 & f(x_m) &= -0.396 \\ x_{m'} &= 0.902 & f(x_{m'}) &= -0.391 \end{aligned}$$

   and :

   $$f(x_{m'}) - f(x_m) = 0.005 < 2^{-0.3*8} = 0.189$$

   Thus we must try again with an higher resolution. Indeed, as can be seen on figure 5, the obtained minimum is clearly sub-optimal.

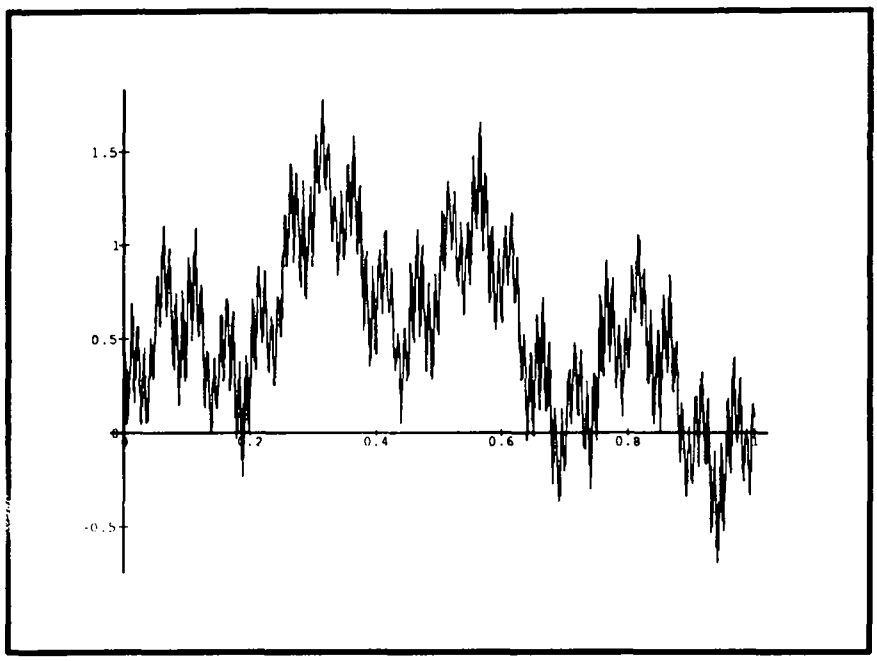

Figure 5: Weierstrass function of dimension 1.7.

With a 28 bits discretization, we obtain :

$$x_m = 0.942477 \qquad f(x_m) = -0.788027$$
$$x_{m'} = 0.942462 \qquad f(x_{m'}) = -0.765587$$

and

$$f(x_{m'}) - f(x_m) = 0.02244 > 2^{-0.3*28} = 0.0029$$

A look at figure 5 confirm that the G.A. has given a good localization.

5 INVERSE PROBLEM FOR IFS

5.1 Recalls on the theory of IFS

We briefly recall the main results of IFS theory. Good references for this are [15, 3, 4], from which most of this paragraph is taken.

Let K be a compact metric space, H be the subset of non-empty closed subsets of K. H is compact with the Hausdorff metric :

$$\forall A \subset K, \forall B \subset K, h(A,B) = max[\sup_{x \in A} \inf_{y \in B} d(x,y), \sup_{x \in B} \inf_{y \in A} d(x,y)]$$

Let $(w_n)_{n=1,\ldots,N}$ be N continuous functions on K.
$\{K, (w_n)_{n=1,\ldots,N}\}$ is called an IFS.

We consider the following function :

$$\begin{aligned} W \quad : \quad & H \to H \\ & A \mapsto \bigcup_n w_n(A) \end{aligned}$$

We call any set $G \subset H$ such that :

$$W(G) = G$$

an attractor of the IFS.

Let S be any subset of H. Then the closure of the set of all accumulation points of $\{W^{\circ n}(S)\}_{n=1}^{\infty}$ is an attractor of $\{K, (w_n)_{n=1,\ldots,N}\}$ (we note $W^{\circ n}(S) = W(W^{\circ(n-1)}(S))$).

Here, we will restrict ourselves to hyperbolic IFS. An IFS is hyperbolic if its functions (w_n) are contractions :

$$\forall n, \exists \lambda_n \in [0,1[/ \forall (x,y) \in K^2, d(w_n(x), w_n(y)) \leq \lambda_n d(x,y)$$

In this case, W is itself a contraction, which implies that the attractor G is unique.

Let us attach to each w_n a probability p_n, such that :

$$\begin{aligned} \forall n, \quad & p_n \geq 0 \\ & \sum_{n=1}^{N} p_n = 1 \end{aligned}$$

A simple method to compute G is then to start from the fixed point x_0 of any w_n, choose at random with the probabilities (p_n) a contraction w_k, compute the image x_1 of x_0 by w_k, and iterate an infinite number of times to obtain a sequence (x_m). Then G is almost surely the set of points x such that any open set containing x also contains infinitely many x_m's.

The definition of the p_n's also induces a probability measure μ with the following property :

for $x \in K$ and B a Borel subset of K, set :

$$P(x, B) = \sum_n p_n \delta_{w_n(x)}(B)$$

where $\delta_y(B) = 0$ if $y \notin B$ and 1 otherwise.

P(x,B) is the probability of transfer from x to B, and thus defines a discrete time Markov process. Then μ is the unique probability measure on K which is stationary for this Markov process. Note that (for hyperbolic IFS) the support of μ is G, independent of the p_n's, as long as they all are strictly positive.

Furthermore, μ is attractive in the following sense :

let ν_0 be any probability measure on K, and define a sequence of probability measures by :

$$\nu_m(B) = \sum_n p_n \nu_{m-1}(w_n^{-1}(B))$$

then $(\nu_m)_m$ converges to μ.

Thus we have, denoting by C^0 the set of continuous real valued functions on K :

$$\forall f \in C^0, \int_K f d\mu = \int_K \sum_n p_n f \circ w_n d\mu \qquad (stationnarity)$$

$$\lim_{m \to \infty} \int_K f d\nu_m = \int_K f d\mu \qquad (attractiveness)$$

5.2 Inverse problem for IFS

So far, we know how to compute, for any hyperbolic IFS, its attractor along with its invariant measure.

The inverse problem reads :

Given a subset G of H, and a probability measure μ on G, find an IFS and some $(p_n)_n$ whose attractor is G and stationary measure is μ.

This problem have been considered by a number of authors. An exact solution can be found in the particular case where each $w_n(G) \bigcap w_m(G)$ is of zero measure (see [5]). In the general case, no exact solution is known. Usually, the problem is divided into two parts :

1. finding the IFS that generate G,
2. finding the (p_n) that generate μ.

An essential tool for solving 1. is the collage theorem :
Let $E \subset K$ be such that :

$$h(E, \bigcup_n w_n(E)) < \epsilon$$

Then :

$$h(A, E) < \frac{\epsilon}{1 - \lambda}$$

where $\lambda = \sup_n\{\lambda_n\}$, and A is the attractor of $(K, (w_n))$

Based on this, several optimization methods have been proposed (see [5, 18, 22, 10, 19].

The resolution of 2. requires the definition of a distance between measures. It has been considered in [22, 15, 3, 24]. An exact solution for a special case, using an extension of the theory presented in [4], is given in [17] : the authors in [4] allow the probabilities to be functions instead of numbers ("place-dependant probabilities"), thus permitting a larger variety of measures to be generated. Conversely, this generalization makes the inverse problem easier to solve in some special cases.

In the rest of this paper, we will restrict ourselves to the resolution of 1.[1].

In [22], the authors propose a G.A. for solving the 1D inverse problem. Their algorithm proves to be efficient on the presented examples. However, most of their choices concerning the parameters of the G.A. itself seems to be ad-hoc, as for instance the size of the population, its evolution, the mutation and mating rules, etc ... Here, we do not pretend to justify all our choices, but we present an algorithm that solves the 2D inverse problem, using the results of section 2 to fix the values of some of the parameters. Note that the 2D problem is much harder than the 1D one, since it involves many more unknowns. Typically, 6 unknowns are present in [22], and 24 in our work.

Our problem can be precisely stated as follows :

Given a compact subset G of the plane, find n contractions (w_i) such that the attractor of (w_i) approximates at best G, in the sense of the Hausdorff metric

Note that we only require that the attractor approximates G, since, in general, there might exist no exact solution. The first thing to compute is the number n of contractions. If the fractal dimension of G in D and if $w_i(G) \bigcap w_j(G)$ is of zero measure for each (i,j), then we know that :

$$\sum_{i=1}^{n} \lambda_i^D = 1$$

[1]The image compression method using IFS does not solve this inverse problem, but a different one, based on the separation of the image into small blocks and the finding of optimal mappings between blocks of different sizes. We just mention if for completeness.

Thus if we ask that all λ_i be lower than λ, we have a lower bound n^* for n, in the special case of non overlapping $w_i(G)$. n^* will also be a lower bound for the general case. In practice, we may compute the fractal dimension of G, and start our investigations with n^* functions. If we are not satisfied with the result we increase the number of functions.

Another unsolved problem is the choice of the basis of functions. Most people use affine transforms for simplicity, but the use of other types of functions, as for instance sine polynomials (see [18] and figure 6) can lead to a much richer class of shapes.

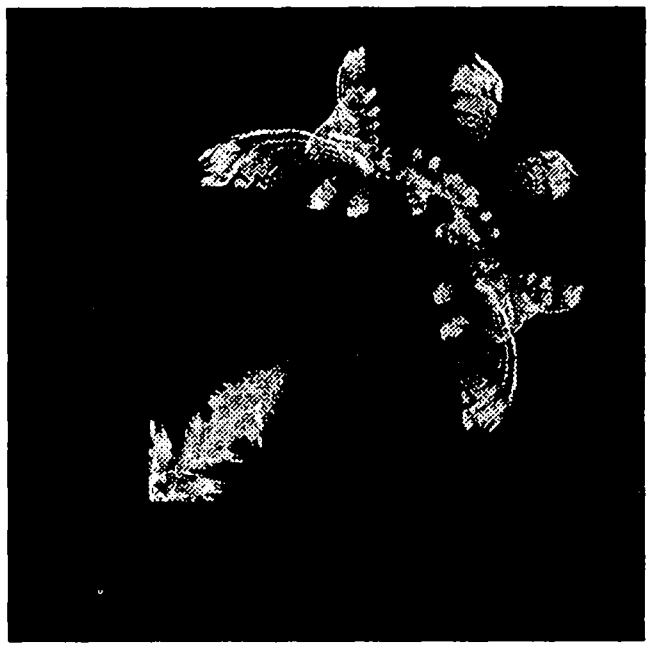

Figure 6: Sine polynomials IFS

Here, we shall restrict ourselves to affine functions. We now proceed to describe the main aspects of the G.A. itself. In order to have an initial idea of the number of needed genes, or equivalently, of the sampling rate of the unknowns, we compute the local scaling exponents of the error function, namely the Hausdorff distance between the attractor of (w_n) and the attractor of $(w_n + \epsilon_n)$ where ϵ_n is a small perturbation. A representation of a 2D slice of this function is displayed on figure 7. We can see that it is extremely irregular, thus indicating that the result of section 2 can be used to evaluate the optimization (although no rigorous demonstration of the fractal structure of the error function has been done).

We have conducted experiments on two attractors : the well-known fern ([6]), and a perturbated Sierpinski triangle (figures 8,9).

The average exponent h estimated on the error function is 0.6, with an incertitude of 0.1. We choose a sampling on 16 bits. The criterion defined in section 2 reads :

$$f(x_{m'}) - f(x_m) > 2^{-8}$$

for normalized values of the different quantities.

This lower bound is sufficiently small to ensure with reasonable probability that the real minimum will be well localized. The results of optimization with our G.A. are on the right of figures 8 and 9.

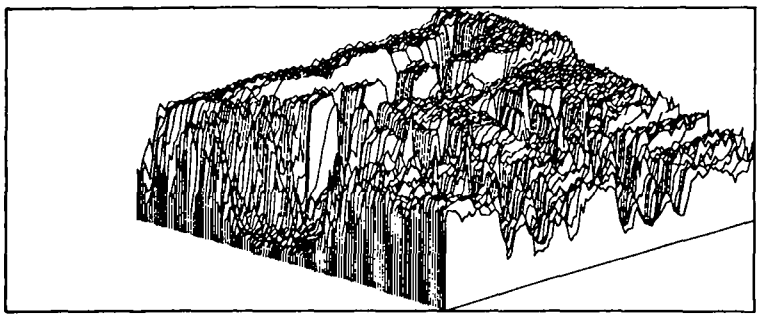

Figure 7: A 2D slice of the error function

The final relative error, in terms of the distance function, is 0.031 for the first attractor, and 0.062 for the second one.

Figure 8: (left) Original 128x128 image : fern - (right) Reconstruction

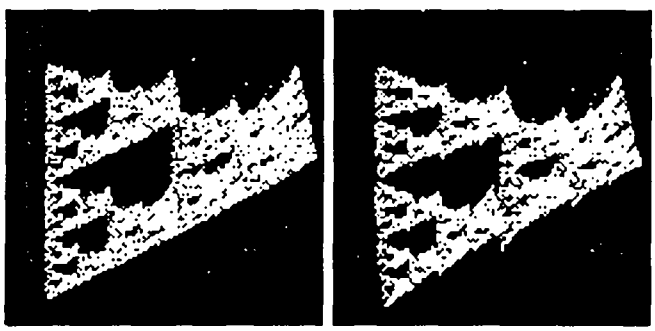

Figure 9: (left) Original 128x128 image : perturbated Sierpinski triangle - (right) Reconstruction

6 CONCLUSION

In this work, we have presented some general results concerning the optimization of fractal functions, in particular relating the local scaling exponents, the sampling rate, and the accuracy of the global minimum localization. We then have used these ideas to design a G.A. for solving the difficult 2D inverse problem for IFS. On the tests we made, the algorithm has been able to find the optimal solution to within an accuracy lower than 6 %, the computing time being of the order of 4 hours.

Much more work is needed to progress towards a general solution of both the optimization of fractal functions and the IFS inverse problem. The bounds of section 2 should be established in a rigorous way, and it is desirable to take into account the integrability of the power spectrum in order to obtain more precise results. As for IFS, we have not tackled the inverse problem for measures, nor approached the use of other types of functions, which are necessary for generating a richer class of shapes.

References

[1] E. Aarts and P. Van Laarhoven. Simulated annealing : a pedestrian review of the theory and some applications. AI Series F30, NATO.

[2] J. Albert, F. Ferri, J. Domingo, and M. Vincens. An approach to natural scene segmentation by means of genetic algoritms with fuzzy data. In *Pattern Recognition and Image Analysis*, pages 97–113, 1992. Selected papers of the 4th Spanish Symposium (Sept 90), Perez de la Blanca Ed.

[3] M. Barnsley and S. Demko. Iterated function system and the global construction of fractals. *Proceedings of the Royal Society*, A 399:243–245, 1985.

[4] M. Barnsley, S. Demko, J. Elton, and Geronimo J. Invariant measures for markov processes arising from iterated function systems with place-dependent probabilities. Georgia Tech. preprint.

[5] M. Barnsley, V. Ervin, D. Hardin, and J. Lancaster. Solution of an inverse problem for fractals and other sets. *Proc. Natl. Acad. Sci. USA*, 83, 1986.

[6] M. F. Barnsley. *Fractals Everywhere.* Academic Press,N Y, 1988.

[7] A. Bihian C. Lemarechal, J.J. Strodiot. *On a bundle algorithm for nonsmooth optimization*, pages 245–282. Academic Press, 1881. Non-Linear Programming 4, Mangasarian, Meyer, Robinson, Editeurs.

[8] Y. Davidor. *Genetic Algorithms and Robotics. A Heuristic Strategy for Optimization.* World Scientific, 1991.

[9] L. Davis. *Genetic Algorithms and Simulated Annealing.* Pittman, London, 1987.

[10] S. Demko, L. Hodges, and B. Naylor. Construction of fractal objects with ifs. *Siggraph*, 19(D 3), 1985.

[11] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):712–741, November 1984.

[12] D. A. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, January 1989.

[13] A. Hill and C. J. Taylor. Model-based image interpretation using genetic algorithms. *Image and Vision Computing*, 10(5):295–301, June 1992.

[14] J. H. Holland. *Adaptation in Natural and Artificial System.* Ann Arbor, University of Michigan Press, 1975.

[15] J. Hutchinson. Fractals and self-similarity. *Indiana University Journal of Mathematics*, 30:713–747; 1981.

[16] J. R. Koza. *Genetic Programming.* MIT Press, 1992.

[17] J. Levy Vehel. Fractal probability functions : an application to image analysis. In *CVPR*, 1991.

[18] J. Levy Vehel and A. Gagalowicz. Fractal approximation of 2d objects. In *EUROGRAPHICS*, 1988.

[19] G. Mantica and A. Sloan. Chaotic optimization and the construction of fractals : solution of an inverse problem. *Complex Systems*, (3):37–62, 1989.

[20] R. Otten and L. Van Ginneken. Annealing : the algorithm. Technical Report RC 10861, March 1984.

[21] G. Roth and M. D. Levine. Geometric primitive extraction using a genetic algorithm. In *IEEE Computer Society Conference on CV and PR*, pages 640–644, 1992.

[22] R. Shonkwiler, F. Mendivil, and A. Deliu. Genetic algorithms for the 1-d fractal inverse problem. In *Genetic Algorithms*, pages 495–502, 1991. 13-16 July.

[23] S. Truvé. Using a genetic algorithm to solve constraint satisfation problems generated by an image interpreter. In *Theory and Applications of Image Analysis : 7th Scandinavian Conference on Image Analysis*, pages 378–386, August 1991. Aalborg (DK).

[24] R. Vrscay. Moment and collage methods for the inverse problem of fractal construction with iterated function systems. In *Fractal 90 Conference*, 1990. Lisbonne, June 6-8.

Unité de Recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

Unité de Recherche INRIA Lorraine Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399